Fishing Club Database Design Project

Project Title

Fishing Club Database Design Project

Brief Description

This project creates a database for a fishing club to track members, trips, catches, events, and locations. The design focuses on clarity, data integrity, and easy reporting, so the club can manage information in a single, consistent system.

Purpose of the Database

The goal is to store all the information in one clean place instead of using messy spreadsheets. The database has seven main tables: Members, Fishing\_Trips, Fish\_Catches, Events, Location, Attendance, and Fish type.

* Members stores people in the club.
* Location keeps addresses for events and trips.
* Fishing\_Trips records each fishing trip.
* Events stores club events like tournaments or meetings.
* Fish\_Catches links to trips and records what fish were caught.

I struggled with creating many-to-many and one-to-one sample data. I was not always sure how to connect the tables correctly. I figured out the one-to-one tables and added a new table for fish types. I think the Fish type table could work as a one-to-one relationship with Fish\_Catches. The overall structure looks good now, but I feel I still need more examples of many-to-many relationships. I also need to better understand which tables should use many-to-many relationships and how to connect them properly. Then I searched for many-to-many relationships to improve my understanding. After that, I connected the Members table to Attendance, then to Events. Because a member can be involved in many events and an event can include many members, Members ↔ Events is many-to-many. That is why I connected them using Attendance as a bridge (junction) table. Now I have a database with a clear many-to-many relationship.

Technologies Used

This project is designed with MySQL in mind as the target database technology. Development and testing are done in MySQL Workbench.

Instructions to Run SQL Scripts

The database currently runs locally on my workstation using MySQL. All table creation and data insert scripts are included.

Task Completion: Tables, Attributes, Primary Keys, Foreign Keys, and Relationships

1) Tables and Attributes (with PK/FK labels)

a) Members

* PK: MemberID (INT, NOT NULL)
* FirstName (VARCHAR)
* LastName (VARCHAR)
* Phone (VARCHAR)
* Email (VARCHAR)
* JoinDate (DATE)
* MemberType (TINYINT)

b) Location

* PK: LocationID (INT, NOT NULL)
* Name (VARCHAR)
* Address (VARCHAR)
* City (VARCHAR)
* State (VARCHAR)
* Zipcode (VARCHAR)

c) Fishing\_Trips

* PK: TripID (INT, NOT NULL, AUTO\_INCREMENT)
* TripDate (DATE, NOT NULL)
* TripTime (TIME, NOT NULL)
* Trip\_Type (TINYINT)
* FK: Location\_LocationID → Location(LocationID)

d) Events

* PK: EventID (INT, NOT NULL, AUTO\_INCREMENT)
* Name (VARCHAR)
* EventsDate (DATE)
* Purpose (VARCHAR)
* FK: Location\_LocationID → Location(LocationID)

e) Fish\_Catches

* PK: CatchID (INT, NOT NULL, AUTO\_INCREMENT)
* Weight (VARCHAR, NOT NULL)
* Length (VARCHAR, NOT NULL)
* Released (TINYINT)
* FK: Fishing\_Trips\_TripID → Fishing\_Trips(TripID)

f) Attendance (junction table for Members ↔ Events)

* PK: AttendanceID (INT, NOT NULL, AUTO\_INCREMENT)
* Role (VARCHAR)
* Type (VARCHAR)
* FK: Members\_MemberID → Members(MemberID)
* FK: Events\_EventID → Events(EventID)

g) Fish type (1:1 with Fish\_Catches in current design)

* PK: TypeID (INT, NOT NULL, AUTO\_INCREMENT)
* SpeciesName (VARCHAR, NOT NULL)
* WaterType (VARCHAR)
* Fish typecol (VARCHAR)
* FK: Fish\_Catches\_CatchID → Fish\_Catches(CatchID)

2) Relationships (with cardinality)

* Location → Fishing\_Trips: 1:M  
  One location can host many trips; each trip occurs at one location.
* Location → Events: 1:M  
  One location can host many events; each event occurs at one location.
* Fishing\_Trips → Fish\_Catches: 1:M  
  One trip can have many catches; each catch belongs to one trip.
* Members ↔ Events (via Attendance): M:M  
  A member can attend many events, and an event can include many members.  
  Bridge: Attendance(Members\_MemberID, Events\_EventID)
* Fish\_Catches ↔ Fish type: 1:1  
  Each catch has exactly one fish type row.